C++ PROGRAMS FOR B.TECH.

Q.1 Write a program to find average of two numbers.

```
//Average of two numbers
#include<iostream>
#include<conio.h>
using namespace std;
int main(void)
{
        int a,b;
        cout<<"Enter 2 numbers";
        cin>>a>>b;
        float avg = float(a+b)/2;
        cout<<"Average is " <<avg <<endl;
        getch();
        return 0;
}
```

Q.2 Write a program to show the use of :: Scope Resolution Operator.

```
// :: Scope Resolution Operator
//To print global variable
#include<iostream.h>
#include<conio.h>
int a=5;  //global variable
void f1();
int main(void)
{
        clrscr();
        cout<<a<<endl;
        int a=10; //local global
        cout<<a<<endl;
        cout<<::a<<endl;
        {
                int a=15;
                cout<<a<<endl;
                cout<<::a<<endl;
                //cout<<::::a;   error
        }
        cout<<a<<endl;
        cout<<::a<<endl;
        f1();
        getch();
        return 0;
}
```

```
void f1()
{
	cout<<a<<endl;
}
```

Q.3 Write a program to show the difference between call by value and call by reference.

```
//Difference between call by value, call by address,
call by reference
#include<iostream.h>
#include<conio.h>
void swapv(int,int);
void swapa(int *,int *);
void swapr(int &,int &);
int main(void)
{
	clrscr();
	int a,b;
	cout<<"Enter two numbers: ";
	cin>>a>>b;	//a=5  b=6
	swapv(a,b);
	cout<<"a="<<a<<"\t" <<"b="<<b<<endl;	//a=5 b=6
	swapa(&a,&b);
	cout<<"a="<<a<<"\t" <<"b="<<b<<endl;	//a=6 b=5
	swapr(a,b);
	cout<<"a="<<a<<"\t" <<"b="<<b<<endl;	//a=5 b=6
	getch();
	return 0;
}
void swapv(int x,int y)
{
	int t=x;x=y;y=t;
	cout<<"x="<<x<<"\t" <<"y="<<y<<endl;	//x=6 y=5
}
void swapa(int *p1,int *p2)
{
	int t=*p1;*p1=*p2;*p2=t;
}
void swapr(int &r1,int&r2)
{
	int t=r1;r1=r2;r2=t;
}
```

Q.4 Write a program to show the use of call by reference.

```
//Example of call by reference
#include<iostream.h>
#include<conio.h>
void sum_avg(int, int, int&,float&);
void main()
{
        int a,b,sum;
        float avg;
        cout<<"Enter two numbers: ";
        cin>>a>>b;
        sum_avg(a,b,sum,avg);
        cout<<"Sum= "<<sum<<endl
            <<"Average= "<<avg;
        getch();
}
void sum_avg(int x,int y,int &r1,float &r2)
{
        r1=x+y;
        r2=(float)r1/2;
}
```

Q.5 Write a program to show the use of function overloading.

```
//Use of function overloading
#include<iostream.h>
#include<conio.h>
void swap(int &, int &);
void swap(float &, float &);
int main(void)
{
        clrscr();
        int a1,b1;
        cout<<"Enter 2 integer values";
        cin>>a1>>b1;
        swap(a1,b1);
        cout<<a1<<"\t"<<b1<<endl;

        float a2,b2;
        cout<<"Enter 2 floating values";
        cin>>a2>>b2;
        swap(a2,b2);
        cout<<a2<<"\t"<<b2<<endl;
        getch();
```

```
	return(0);
}
void swap(int &r1, int &r2)
{
	int t=r1;r1=r2;r2=t;
}
void swap(float &r1, float &r2)
{
	float t=r1;r1=r2;r2=t;
}
```

Q.6 Write a program to show the use of Default Arguments.

```
//Default Arguments
#include<iostream.h>
#include<conio.h>
void print(char='*', int=10);
void main()
{
	clrscr();
	print('+',5);
	print('-');
	print();
	// print( ,7);  error
	getch();
}
void print(char ch, int n)
{
	for(int i=1;i<=n;i++)
		cout<<ch;
	cout<<endl;
}
```

Q.7 Write a program to swap of two numbers.

```
#include<iostream.h>
#include<conio.h>
inline void swap(int &r1,int &r2)
{
	int t=r1;
	r1=r2;
	r2=t;
}
int main(void)
{
	clrscr();
```

```
        int a1,b1;
        cout<<"Enter two numbers: ";
        cin>>a1>>b1;
        swap(a1,b1);
        cout<<a1<<"\t"<<b1<<endl;
        int a2,b2;
        cout<<"Enter two numbers: ";
        cin>>a2>>b2;
        swap(a2,b2);
        cout<<a2<<"\t" <<b2<<endl;
        int a3,b3;
        cout<<"Enter two numbers: ";
        cin>>a3>>b3;
        swap(a3,b3);
        cout<<a3<<"\t" <<b3<<endl;
        getch();
        return 0;
}
```

Q.8 Write a program to show the use of Macro Function.

```
//Macro function
#include<iostream.h>
#include<conio.h>
#define sqr(a) a*a
#define max(a,b) a>b?a:b;
void main()
{
        clrscr();
        int x1,y1,ans1;
        cout<<"Enter 2 numbers";
        cin>>x1>>y1;
        ans1=max(x1,y1);
        cout<<"Max is "<<ans1<<endl;
        int n,ans2;
        cout<<"Enter a number";
        cin>>n;
        ans2=sqr(n);
        cout<<"Square is "<<ans2;
        getch();
}

/*Side affects of Macro
1.      No Type Checking
        inline int mod(int a, int b)
        {
            return a%b;
        }
        main()
        {
            cout<<mod(5.0,2.0);   there will be implicit typecasting to int
        }

        #define mod(a,b) a%b
        main()
        {
            cout<<mod(5.0,2.0);   error
        }
```

2. Bug hunting is difficult in macro

```
        #define max(a,b) a>b?a?b
        main()
        {
            cout<<max(a1,b1);   error
            cout<<max(a2,b2);   error
            cout<<max(a3,b2);   error
        }
```

3. Increment Decrement side affect

```
        inline int sqr(int a)
        {
            return a*a;
        }
        main()
        {
            int x=5;
            cout<<sqr(++x);         //36
        }

        #define sqr(a) a*a
        main()
        {
            int x=5;
            cout<<sqr(++x);
                    //42 but in old compilers 49
        }

*/
```

Q.9 Write a program to sort a one dimensional array using selection sorting method.

```
//selection sorting in dynamic 1-dimensional array
#include<iostream.h>
#include<stdlib.h>
#include<conio.h>
int main(void)
{
	clrscr();
	int *p,n;
	cout<<"Enter number of elements: ";
	cin>>n;

	p=new int[n];
	if(p==NULL)
	{
		cerr<<"Memory is full";
		getch();
		exit(1);
	}
	//input
	for(int i=0;i<n;i++)
	{
		cout<<"Enter number "<<i+1 << " : ";
		cin>>p[i];
	}
	//sorting
	for(i=0;i<n-1;i++)
	{
		for(int j=i+1;j<n;j++)
		{
			if(p[i]>p[j])
			{
				int t=p[i];
				p[i]=p[j];
				p[j]=t;
			}
		}
	}
	//output
	for(i=0;i<n;i++)
		cout<<p[i]<<endl;

	delete [] p;
	getch();
```

```
        return 0;
}
```

Q.10 Write a program to transpose a matrix in two dimensional array.

```
//matrix transpose in 2 dimensional array
#include<iostream.h>
#include<conio.h>
int main(void)
{
        int **p,r,c;
        clrscr();
        cout<<"Enter number of rows ";
        cin>>r;
        cout<<"Enter number of columns ";
        cin>>c;

        p=new int*[r];
        for(int i=0;i<r;i++)
        {
                p[i]=new int[c];
        }

        //input
        for(i=0;i<r;i++)
        {
                for(int j=0;j<c;j++)
                {
                        cout<<"Enter element "<<i+1<<","<<j+1<<": ";
                        cin>>p[i][j];
                }
        }
        //output
        cout<<"original matrix"<<endl;
        for(i=0;i<r;i++)
        {
                for(int j=0;j<c;j++)
                {
                        cout<<p[i][j]<<"\t";
                }
                cout<<endl;
        }
        //transpose
        cout<<"transposed matrix"<<endl;
        for(i=0;i<c;i++)
        {
                for(int j=0;j<r;j++)
                {
                        cout<<p[j][i]<<"\t";
                }
                cout<<endl;
        }
        //free memory
        for(i=0;i<r;i++)
        {
                delete  []   p[i];
        }
        delete  [] p;
        getch();
        return 0;
}
```

Q.11 Write a program to show the use of three dimensional array.

```
//matrix 3 dimensional array
#include<iostream.h>
#include<conio.h>
int main(void)
{
        int ***p,m,r,c;
        clrscr();
        cout<<"Enter number of matrix ";
        cin>>m;
        cout<<"Enter number of rows ";
        cin>>r;
        cout<<"Enter number of columns ";
        cin>>c;

        p=new int **[m];
        for(int i=0;i<m;i++)
        {
                p[i]=new int*[r];
                for(int j=0;j<r;j++)
                {
                        p[i][j]=new int[c];
                }
        }

        //input
        for(i=0;i<m;i++)
          for(int j=0;j<r;j++)
            for(int k=0;k<c;k++)
            {
                cout<<"Enter element "<<i+1<<","<<j+1<<","<<k+1<<": ";
                cin>>p[i][j][k];
            }

        //output
        cout<<"original matrix"<<endl;
        for(i=0;i<m;i++)
        {
          for(int j=0;j<r;j++)
          {
            for(int k=0;k<c;k++)
                cout<<p[i][j][k]<<"\t";
            cout<<endl;
          }
          cout<<endl;
        }
        for(i=0;i<m;i++)
        {
                for(j=0;j<r;j++)
                        delete [] p[i][j];
                delete [] p[i];
        }
        delete [] p;
        getch();
        return 0;
}
```

Q.12 Write a program to show the use of pointers.

```
#include<iostream.h>
#include<conio.h>
int main(void)
{
    int a=5;
    float b=5.4f;

    int *p1;
    p1=&a;
    *p1=10;
    cout<<"a=" <<a<<endl;
    p1++;
//  p1=&b; error


    float *p2;
```

```
        p2=&b;
        *p2=6.3f;
        cout<<"b="<<b<<endl;
        p2++;
//      p2=&a; error

        void *p3;

        p3=&a;
//      *p3=15; error
        *(int *)p3=15;
        cout<<"a="<<a<<endl;
//      p3++; error
        ((int *)p3)++;

        p3=&b;
//      *p3=7.2f; error
        *(float *)p3=7.2f;
        cout<<"b="<<b<<endl;
//      p3++; error
        ((float *)p3)++;

        getch();
        return 0;
}
```

//Rules of reference

1. Reference variable must be declared & initialized simultaneously

```
        int a=5;                    int a=5;
        int &r=a;    valid            int &r;    invalid
                                          r=a;
```

but Pointer can be declared & initialized seperatly

```
        int a=5;                    int a=5;
        int *p=&a;    valid           int *p;        valid
                                          p=&a;
```

2. Once a reference variable is tied to a variable it can not be tied to some other variable

```
        int a=5,b=6;
        int &r=a;
        r=10;
        int &r=b;   //error
        r=20;
```

```
A pointer can store address of one variable at a time but
it can store the address of other variable at some other time
int a=5, b=6;
int *p;
p=&a;
*p=10;
p=&b;     //no error
*p=20;
```

3. There can be more than one reference of a single variable

```
int a=5;
int &r1=a;
r1=10;
int &r2=a;
r2=20;
cout<<a<<r1<<r2;     //20 20 20
```

4. Array of pointers is possible but Array of References is not possible

```
int a,b,c,d,e;
int *p[5]={&a,&b,&c,&d,&e};  //valid
int &r[5]={a,b,c,d,e};   //not valid
```

//Constant

1. Constant value

```
float pi=3.141f;
pi=50;
3.141f=50;  //can't change a constant
```

2. Constant Variable

```
const float PI=3.141f;
PI=50; //can't change a constant
```

3. Symbolic Constant

```
#define PI 3.141
PI=50; //can't change a constant
```

4. Constant Pointer

a.
```
char *p="abc";
cout<<p<<endl;
*p='z';          //string changed
cout<<p<<endl;
```

```
        p="xyz";        //pointer changed
        cout<<p<<endl;

b.      char * const p="abc";   //pointer is constant
        cout<<p<<endl;
        *p='z';         //string changed
        cout<<p<<endl;
        p="xyz";        //error can't change the pointer

c.      const char *p="abc"; //string is constant
        cout<<p<<endl;
        *p='z';         //error can't change the string
        p="xyz";        //pointer changed
        cout<<p<<endl;

d.      char const *p="abc"; //string is constant
        cout<<p<<endl;
        *p='z';         //error can't change the string
        p="xyz";        //pointer changed
        cout<<p<<endl;
```

5. Constant Reference

```
a.      const int a=5;
        int &r=a;
        a=10;           //error can't change a constant
        r=10;           //no error as r is not a constant but a is constant
                        // so compiler will allocate seperate bytes
                        // to r, now r is not the reference of a.
        cout<<a<<r;     //5     10

b.      int a=5;
        const int &r=a;
        r=10;           //error can't change a constant
        a=10;           //no error as a is not a constant but r is constant
                        // but r is only a reference of a so r has to
                        // accept the new value.
        cout<<a<<r;     //10    10
```

Q.13 Write a program of Complex Number.

```
#include<iostream.h>
#include<conio.h>
struct complex
{
        int real;
```

```
	int imag;
};
void getdata(complex &);
void display(complex);
complex sum(complex,complex);
//or
//complex operator +(complex,complex);

complex mult(complex, complex);
//or
//complex operator *(complex, complex);

void main()
{
	clrscr();
	complex c1,c2,c3,c4;
	getdata(c1);
	getdata(c2);
	c3=sum(c1,c2);
	//or
	//c3=c1+c2;
	cout<<"sum is ";
	display(c3);
	c4=mult(c1,c2);
	//or
	//c4=c1+c2;
	cout<<"product is ";
	display(c4);
	getch();
}
void getdata(complex &c)
{
	cout<<"Enter real number: ";
	cin>>c.real;
	cout<<"Enter imaginary number: ";
	cin>>c.imag;
}
void display(complex c)
{
	if(c.imag>=0)
		cout<<c.real<<"+"<<c.imag<<"i"<<endl;
	else
		cout<<c.real<<c.imag<<"i"<<endl;
}
//complex operator +(complex c1, complex c2)
//or
```

```
complex sum(complex c1, complex c2)
{
        complex t;
        t.real=c1.real+c2.real;
        t.imag=c1.imag+c2.imag;
        return(t);
}
//complex operator *(complex c1, complex c2)
//or
complex mult(complex c1, complex c2)
{
        complex t;
        t.real=c1.real*c2.real-c1.imag*c2.imag;
        t.imag=c1.real*c2.imag+c1.imag*c2.real;
        return(t);
}
```

Q.14 Write a program to add and multiply of two matrices.

```
//add and multiply two matrices
#include<iostream.h>
#include<conio.h>
#include<process.h>

#define ROW 10
#define COL 10

struct matrix
{
        int mat[ROW][COL];
        int r,c;
};
void getdata(matrix &);
void display(matrix m);
//matrix operator+(matrix, matrix);
//or
matrix sum(matrix, matrix);

//matrix operator*(matrix, matrix);
//or
matrix mult(matrix, matrix);
int main(void)
{
        matrix m1,m2,m3;
        getdata(m1);
        getdata(m2);
```

```
	//m3=m1+m2;
	//or
	m3=sum(m1,m2);
	cout<<"Sum is "<<endl;
	display(m3);
	matrix m4,m5,m6;
	getdata(m4);
	getdata(m5);
	//m6=m4*m5;
	//or
	m6=mult(m4,m5);
	cout<<"Product is "<<endl;
	display(m6);
	getch();
	return 0;
}
void getdata(matrix &m)
{
	cout<<"Enter number of rows";
	cin>>m.r;
	cout<<"Enter number of columns";
	cin>>m.c;
	for(int i=0;i<m.r;i++)
	{
		for(int j=0;j<m.c;j++)
		{
			cout<<"Enter element"<<i+1<<","<<j+1<<" ";
			cin>>m.mat[i][j];
		}
	}
}
void display(matrix m)
{
	for(int i=0;i<m.r;i++)
	{
		for(int j=0;j<m.c;j++)
			cout<<m.mat[i][j]<<"\t";
		cout<<endl;
	}
}
//matrix operator+ (matrix m1,matrix m2)
//or
matrix sum(matrix m1,matrix m2)
{
	if(m1.r!=m2.r||m1.c!=m2.c)
	{
```

```
                cout<<"cannot add";
                exit(1);
        }
        matrix t;
        t.r=m1.r;
        t.c=m1.c;
        for(int i=0;i<m1.r;i++)
        {
                for(int j=0;j<m1.c;j++)
                        t.mat[i][j]=m1.mat[i][j]+m2.mat[i][j];
        }
        return(t);
}
//matrix operator* (matrix m1,matrix m2)
//or
matrix mult(matrix m1,matrix m2)
{
        if(m1.c!=m2.r)
        {
                cout<<"cannot multiply";
                exit(1);
        }
        matrix t;
        t.r=m1.r;
        t.c=m2.c;
        for(int i=0;i<m1.r;i++)
        {
                for(int j=0;j<m2.c;j++)
                {
                        t.mat[i][j]=0;
                        for(int k=0;k<m2.r;k++)
                                t.mat[i][j]+=m1.mat[i][k]*m2.mat[k][j];
                }
        }
        return(t);
}
```

Q.15 Write a program to show a simple C++ Program using structure.

```
#include<iostream.h>
#include<conio.h>
struct A
{
        int x;
        int y;
};
```

```
void getdata(A &obj)
{
        cout<<"Enter two numbers:";
        cin>>obj.x>>obj.y;
}
void display(A obj)
{
        cout<<obj.x<<"\t"<<obj.y<<endl;
}
void main()
{
        A a1,a2;
        getdata(a1);
        getdata(a2);
        display(a1);
        display(a2);
        getch();
}
```

Q.16 Write a program to show a simple C++ Program using Class.

```
#include<iostream.h>
#include<conio.h>
class A
{
    private:
        int x;
        int y;
    public:
        void getdata()
        {
                cout<<"Enter two numbers:";
                cin>>x>>y;
        }
        void display()
        {
                cout<<x<<"\t"<<y<<endl;
        }
};
void main()
{
        A a1,a2;
        a1.getdata();
        a2.getdata();
        a1.display();
        a2.display();
        getch();
}
```

Q.17 Write a program to show student information using class.

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class student
{
        int roll;
        char name[10];
        int sub1,sub2;
        int total()
        {
                return(sub1+sub2);
        }
        float per()
        {
                int t=total();
                return (float)t/2;
        }
    public:
        void getdata()
        {
                cout<<"Enter roll number:";
                cin>>roll;
                char ch;
                ch=cin.get();
                cout<<"Enter name:";
                cin.getline(name,10);
                cout<<"Enter marks in 2 subjects:";
                cin>>sub1>>sub2;
        }
        void setdata(int r,char n[],int a,int b)
        {
                roll=r;
                strcpy(name,n);
                sub1=a;
                sub2=b;
        }
        void display()
        {
           cout<<roll<<"\t"<<name<<"\t"<<sub1<<"\t"<<sub2<<"\t"<<per()<<endl;
        }
```

```
};
int main(void)
{
        student s1,s2;
        s1.getdata();
        s2.getdata();
        student s3,s4;
        s3.setdata(7,"amit",78,90);
        s4.setdata(8,"neha",87,66);
        cout<<"roll\tname\tsub1\tsub2\tper\n";
        s1.display();
        s2.display();
        s3.display();
        s4.display();
        getch();
        return 0;
}
//or
/*
int main(void)
{
        student s[10];
        int n;
        cout<<"Enter how many students";
        cin>>n;
        for(i=0;i<n;i++)
                s[i].getdata();
        for(i=0;i<n;i++)
                s[i].display();
        getch();
        return 0;
}
*/
//or
/*
int main(void)
{
        student *s;
        int n;
        cout<<"Enter how many students";
        cin>>n;
        s=new student[n];
        for(i=0;i<n;i++)
                s[i].getdata();
        for(i=0;i<n;i++)
                s[i].display();
        delete []s;
        getch();
        return 0;
}
*/
```

Q.18 Write a program to show the use of fflush() function in C.

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int roll;
        char name[20];
        clrscr();
        printf("Enter a roll");
        scanf("%d",&roll);
        fflush(stdin);
        printf("Enter a name");
                //scanf("%s",name);
        gets(name);

        printf("\n%d\t%s",roll,name);
        getch();
}
```

Q.18 Write a program to show flush buffer data without using fflush() function in C++.

```
#include<iostream.h>
#include<conio.h>
void main()
{
        clrscr();
        int roll;
        char name[20];

        cout<< "Enter a roll";
        cin>>roll;

        cout<<"Enter a name";
                //cin>>name;
        char ch;
        ch=cin.get();
        cin.getline(name,20);

        cout<<endl<<roll<<"\t" <<name;
        getch();
}
```

Q.19 Write a program to show the use of THIS Pointer.

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class student
{
        int roll;
        char name[10];
        int sub1,sub2;
        int total()
        {
                return(sub1+sub2);
        }
        float per()
        {
                int t=total();
                return (float)t/2;
        }
    public:
        void getdata()
        {
                cout<<"Enter roll number:";
                cin>>roll;        //implicit use of this
                char ch;
                ch=cin.get();
                cout<<"Enter name:";
                cin.getline(name,10);
                cout<<"Enter marks in 2 subjects:";
                cin>>sub1>>sub2;
        }
        void setdata(int roll,char name[],int sub1,int sub2)
        {
                this->roll=roll;   //explicit use of this
                strcpy(this->name,name);
                this->sub1=sub1;
                this->sub2=sub2;
        }
        void display()
        {
           cout<<roll<<"\t"<<name<<"\t"<<sub1<<"\t"<<sub2<<"\t"<<per()<<endl;
        }
};
int main(void)
{
        student s1,s2;
        s1.getdata();
        s2.getdata();
        student s3,s4;
        s3.setdata(7,"amit",78,90);
        s4.setdata(8,"neha",87,66);
        cout<<"roll\tname\tsub1\tsub2\tper\n";
        s1.display();
        s2.display();
        s3.display();
        s4.display();
        getch();
        return 0;
}
```

Q.20 Write a program to enter and display of two values using class in C++.

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class A
{
        int x,y;
    public:
        void getdata();
        void display();
};
void A::getdata()
{
        cout<<"Enter two numbers:";
        cin>>x>>y;
}
void A::display()
{
  cout<<x<<"\t"<<y<<endl;
}
int main(void)
{
        A a1;
        a1.getdata();
        a1.display();
        getch();
        return 0;
}
```

Q.21 Write a program to show the use of constructor and destructor in C++.

```
#include<iostream.h>
#include<conio.h>
class A
{
        int x,y;
        public:
        A()//zero argument constructor
        {
                x=y=0;
        }
        A(int x1)//parameterized one argument constructor
        {
                x=y=x1;
        }
        A(int x1, int y1)//parameterized two argument constructor
        {
                x=x1;
                y=y1;
        }
        void getdata()
        {
                cout<<"Enter 2 numbers ";
                cin>>x>>y;
        }
        void display()
        {
                cout<<x<<"\t"<<y<<endl;
        }
        ~A() //destructor
        {
        }
};
int main(void)
{
        clrscr();
        A a1;
        a1.display();
        A a2(5);
        a2.display();
        A a3(5,7);
        a3.display();
        A a4;
        a4.getdata();
        a4.display();
        getch();
        return 0;
}
```

Q.22 Write a program to add and multiply of two complex numbers using class in C++.

```
#include<iostream.h>
#include<conio.h>
class complex
{
        int real,imag;
    public:
        complex();
        complex(int,int);
        void getdata();
        void display();
        complex sum(complex);
        complex mult(complex);
};
complex::complex()
{
        real=imag=0;
}
complex::complex(int real,int imag)
{
        this->real=real;
        this->imag=imag;
}
void complex::getdata()
{
        cout<<"Enter real number";
        cin>>real;
        cout<<"Enter imaginary number";
        cin>>imag;
}
void complex::display()
{
        if(imag>=0)
                cout<<real<<"+"<<imag<<"i"<<endl;
        else
                cout<<real<<imag<<"i"<<endl;
}
complex complex::sum(complex c)
{
        complex t;
        t.real=real+c.real;
        t.imag=imag+c.imag;
        return (t);
//or
//      complex t(real+c.real,imag+c.imag);
//      return (t);
//or
//      return complex(real+c.real,imag+c.imag);
}
complex complex::mult(complex c)
{
        complex t;
        t.real=real *c.real-imag * c.imag;
        t.imag=imag *c.real+real *c.imag;
        return (t);
}
int main(void)
{
        complex c1,c2,c3,c4;
        c1.getdata();
        c2.getdata();
        c3=c1.sum(c2);
        cout<<"Sum =";
        c3.display();
        c4=c1.mult(c2);
        cout<<"Multiply =";
        c4.display();
        getch();
        return 0;
}


/*complex t(real+c.real,imag+c.imag);
return (t);     */
/*complex t= complex(real+c.real, imag+c.imag);
return(t)  */
/*return(complex(real+c.real, imag+c.imag); */
```

Q.23 Write a program to show addition and multiplication of two 2 dimension matrix using class in C++.

```
#include<iostream.h>
#include<conio.h>
#include<process.h>
#define ROW 10
```

```
#define COL 10
class matrix
{
        int r,c;
        int mat[ROW][COL];
    public:
        matrix();
        matrix(int,int);
        void getdata();
        void display();
        matrix sum(matrix);
        matrix mult(matrix);
};
matrix::matrix()
{
        r=c=0;
}
matrix::matrix(int r,int c)
{
        this->r=r;
        this->c=c;
}
void matrix::getdata()
{
        cout<<"Enter number of rows";
        cin>>r;
        cout<<"Enter number of columns";
        cin>>c;
        for(int i=0;i<r;i++)
                for(int j=0;j<c;j++)
                {
                        cout<<"Enter element"<<i+1<<","<<j+1<<" ";
                        cin>>mat[i][j];
                }
}
void matrix::display()
{
        for(int i=0;i<r;i++)
        {
                for(int j=0;j<c;j++)
                        cout<<mat[i][j]<<"\t";
                cout<<endl;
        }
}
matrix matrix::sum(matrix m)
{
        if(r!=m.r||c!=m.c)
        {
                cout<<"cannot add";
                getch();
                exit(1);
        }
        matrix t(r,c);
        for(int i=0;i<r;i++)
        {
                for(int j=0;j<c;j++)
                        t.mat[i][j]=mat[i][j]+m.mat[i][j];
        }
        return(t);
}
matrix matrix::mult(matrix m)
{
        if(c!=m.r)
        {
                cout<<"cannot multiply";
                getch();
                exit(1);
        }
        matrix t(r,m.c);
        for(int i=0;i<r;i++)
                for(int j=0;j<m.c;j++)
                {
                        t.mat[i][j]=0;
                        for(int k=0;k<c;k++)
                                t.mat[i][j]+=mat[i][k]*m.mat[k][j];
                }
        return(t);
}
int main(void)
{
        clrscr();
        matrix m1,m2,m3;
        m1.getdata();
        m2.getdata();
        m3=m1.sum(m2);
        cout<<"sum="<<endl;
        m3.display();
        matrix m4,m5,m6;
        m4.getdata();
        m5.getdata();
        m6=m4.mult(m5);
        cout<<"multiply="<<endl;
        m6.display();
        getch();
        return 0;
}
```

Q.24 Write a program to show addition and multiplication of two 2 dimention matrix using constructor and destructor in C++ with the help of NEW and DELETE function.

```
#include<iostream.h>
#include<conio.h>
#include<process.h>
class matrix
{
        int **mat;
        int r,c;
    public:
        matrix();
        matrix(int,int);
        void getdata();
        void display();
        matrix sum(matrix);
        matrix mult(matrix);
        ~matrix();
};
matrix::matrix()
{
        r=c=0;
        mat=NULL;
}
matrix::matrix(int r,int c)
{
        this->r=r;
        this->c=c;
        mat=new int *[r];
        for(int i=0;i<r;i++)
                mat[i]=new int[c];
}
void matrix::getdata()
{
        int i,j;
        cout<<"Enter number of rows";
        cin>>r;
        cout<<"Enter number of columns";
        cin>>c;
        mat=new int *[r];
```

```
	for(i=0;i<r;i++)
		mat[i]=new int[c];

	for(i=0;i<r;i++)
		for(j=0;j<c;j++)
		{
			cout<<"Enter element"<<i+1<<","<<j+1<<" ";
			cin>>mat[i][j];
		}
}
void matrix::display()
{
	for(int i=0;i<r;i++)
	{
		for(int j=0;j<c;j++)
			cout<<mat[i][j]<<"\t";
		cout<<endl;
	}
}
matrix matrix::sum(matrix m)
{
	if(r!=m.r||c!=m.c)
	{
		cout<<"cannot add";
		getch();
		exit(1);
	}
	matrix t(r,c);
	for(int i=0;i<r;i++)
	{
		for(int j=0;j<c;j++)
			t.mat[i][j]=mat[i][j]+m.mat[i][j];
	}
	return(t);
}
matrix matrix::mult(matrix m)
{
	if(c!=m.r)
	{
		cout<<"cannot multiply";
		getch();
		exit(1);
	}
	matrix t(r,m.c);
	for(int i=0;i<r;i++)
		for(int j=0;j<m.c;j++)
```

```
                {
                        t.mat[i][j]=0;
                        for(int k=0;k<c;k++)
                                t.mat[i][j]+=mat[i][k]*m.mat[k][j];
                }
        return(t);
}
matrix::~matrix()
{
        for(int i=0;i<r;i++)
                delete [] mat[i];
        delete [] mat;
}
int main(void)
{
        clrscr();
        matrix m1,m2,m3;
        m1.getdata();
        m2.getdata();
        m3=m1.sum(m2);
        cout<<"sum="<<endl;
        m3.display();
        matrix m4,m5,m6;
        m4.getdata();
        m5.getdata();
        m6=m4.mult(m5);
        cout<<"multiply="<<endl;
        m6.display();
        getch();
        return 0;
}
```

Q.25 Write a program to show the use of dynamic constructor in C++.

```
//dynamic constructor
#include<iostream.h>
#include<conio.h>
class A
{
        int *p;
        int n;
    public:
        A()
        {
                n=0;
                p=NULL;
```

```
	}
	A(int n1)
	{
		n=n1;
		p=new int [n];
	}
	void getdata();
	void display();
	~A()
	{
		delete []p;
	}
};
void A::getdata()
{
	for(int i=0;i<n;i++)
	{
		cout<<"Enter number "<<i+1;
		cin>>p[i];
	}
}
void A::display()
{
	for(int i=0;i<n;i++)
		cout<<p[i]<<endl;
}
int main (void)
{
	int x;
	cout<<"Enter number of elements: ";
	cin>>x;
	A a1(x);
	a1.getdata();
	a1.display();
	getch();
	return 0;
}
```

Q.26 Write a program to show the use of Copy constructor in C++.

```
//copy constructor
#include<iostream.h>
#include<conio.h>
class A
{
	int x,y;
```

```
        public:
        A()
        {
                x=y=0;
        }
        A(int x1,int y1)
        {
                x=x1;
                y=y1;
        }
        A(A &obj)
        {
                x=obj.x *2;
                y=obj.y *2;
        }
        void display()
        {
                cout<<x<<endl<<y;
        }
};
int main(void)
{
        clrscr();
        A a1(5,6);
        A a2(a1);
//or
//      A a2=a1;
        a2.display();
        getch();
        return 0;
}
```

Q.27 Write a program to show the use of unary operator overloading (increment) in C++.

```
//unary op. overloading (Increment)
#include<iostream>
#include<conio.h>
using namespace std;
class counter
{
        int cnt;
        public:
        counter()
        {
                cnt=0;
        }
        void display()
        {
                cout<<cnt<<endl;
        }
        int operator++()//prefix
        {
                return(++cnt);
        }
        int operator++(int)//postfix
        {
                return(cnt++);
        }
};
int main(void)
{
        counter c1;
        int a=c1++;
        int b=++c1;
        cout<<"a="<<a<<endl;
        cout<<"b="<<b<<endl;
        c1.display();
        getch();
        return 0;
}
```

Q.28 Write a program to show the use of static and instance variables in C++.

```
//static and instance variable
#include<iostream.h>
#include<conio.h>
class A
{
    public:
        int x;
        static int y;//declaration of static variable no memory
};
int A::y;//defination of static variable memory allocation
int main(void)
{
        cout<<A::y<<endl;
        A a1,a2,a3;
        a1.x=10;a1.y=20;
        a2.x=11;a2.y=21;
        a3.x=12;a3.y=22;
        cout<<a1.x<<"\t"<<a1.y<<endl;
        cout<<a2.x<<"\t"<<a2.y<<endl;
        cout<<a3.x<<"\t"<<a3.y<<endl;
        getch();
        return 0;
}
```

Q.29 Write a program to show the use of static and instance variables using constructor and destructor in C++.

```
//static and instance variable
#include<iostream.h>
#include<conio.h>
class counter
{
        static int cnt;//declaration of static variable no memory
        public:
        counter()
        {
                cnt++;
        }
        void display()
        {
//              cout<<cnt<<endl;
        }
        ~counter()
        {
                cnt--;
                cout<<cnt;
        }
};
int counter::cnt;//defination of static variable memory allocation
int main (void)
{
        clrscr();
        counter c1,c2,c3;
        c1.display();
        c2.display();
        c3.display();
        {
                counter c4;
                c1.display();
        }
        c2.display();
        getch();
        return 0;
        //Press Alt+F5 to see the output again.
}
```

Q.30 Write a program to show the use of member functions in C++.

```
//member functions
#include<iostream.h>
#include<conio.h>
class math
{
	public:
	static int sum(int x,int y)
	{
		return (x+y);
	}
	static float avg(int x,int y)
	{
		return(float(x+y)/2);
	}
};
int main()
{
	int x,y;
	cout<<"Enter two numbers";
	cin>>x>>y;
	cout<<"Sum = "<<math::sum(x,y)<<endl;
	cout<<"Average = "<<math::avg(x,y)<<endl;

	math m1;
	cout<<sizeof(m1)<<endl;
	int a,b;
	cout<<"enter two numbers";
	cin>>a>>b;
	cout<<"sum = "<<m1.sum(a,b)<<endl;
	cout<<"average = "<<m1.avg(a,b)<<endl;
	getch();
	return 0;
}
```

Q.31 Write a program to show the use of constant member functions in C++.

```
//const member function
#include<iostream.h>
#include<conio.h>
class A
{
	int x;
	mutable int y;
```

```
    public:
        A()
        {
                x=y=0;
        }
        A(int x1, int y1)
        {
                x=x1;
                y=y1;
        }
        void getdata()
        {
                cout<<"Enter 2 numbers";
                cin>>x>>y;
        }
        void display() const
        {                                               x=10;y=20;
                cout<<x<<"\t"<<y<<endl;
        }
};
int main()
{
        A a1;
        a1.getdata();
        a1.display();
        getch();
        return 0;
}
```

Q.32 Write a program to show the use of constant object in C++.

```
//const object
#include<iostream.h>
#include<conio.h>
class A
{
        int x;
        int y;
    public:
        A()
        {
                x=y=0;
        }
        A(int x1, int y1)
        {
                x=x1;
```

```
		y=y1;
	}
	void getdata()
	{
		cout<<"Enter 2 numbers";
		cin>>x>>y;
	}
	void display() const
	{
		cout<<x<<"\t"<<y<<endl;
	}
};
int main()
{
	const A a1(5,6);
	a1.getdata();	//warning call of non constant
			// function with const object
	a1.display();
	getch();
	return 0;
}
```

Q.33 Write a program to converse object to primitive in C++.

```
//conversion from object to primitive
#include<iostream.h>
#include<conio.h>
class A
{
	int x;
	float y;
	public:
	A()
	{
		x=y=0;
	}
	A(int x1)
	{
		x=x1;
		y=0;
	}
	A(float y1)
	{
		x=0;
		y=y1;
	}
```

```
        void display()
        {
                cout<<endl<<x<<"\t"<<y;
        }
        operator int()
        {
                return x;
        }
        operator float()
        {
                return y;
        }
};
void main()
{
        clrscr();
        A a1,a2;
        a1=5;
        a2=5.3f;
        a1.display();
        a2.display();

        int z1;
        z1=a1;
//or
//      z1=(int)a1;

        float z2;
        z2=a1;
//or
//      z2=(float)a1;

        cout<<endl<<z1<<"\t"<<z2;
        getch();
}
```

Q.34 Write a program to show the conversion function in destination in C++.

```
//conversion function in destination
#include<iostream.h>
#include<conio.h>
class B
{
        int y;
        public:
        B()
```

```
	{
		y=0;
	}
	B(int y1)
	{
		y=y1;
	}
	int gety()
	{
		return(y);
	}
};
class A
{
	int x;
	public:
	A()
	{
		x=0;
	}
	A(B b1)
	{
		x=b1.gety();
	}
	void display()
	{
		cout<<x<<endl;
	}
};
void main()
{
	clrscr();
	A a1;
	B b1(7);
	a1=b1;
	a1.display();
	getch();
}
```

Q.35 Write a program to show the conversion function in source in C++.

```
//conversion function in source
#include<iostream.h>
#include<conio.h>
class A
{
	int x;
	public:
	A()
	{
		x=0;
	}
	A(int x1)
	{
		x=x1;
	}
	void display()
	{
		cout<<x<<endl;
	}
};
class B
{
	int y;
	public:
	B()
	{
		y=0;
	}
	B(int y1)
	{
		y=y1;
	}
	operator A()
	{
		A obj(y);
		return(obj);
	}
};
void main()
{
	clrscr();
	A a1;
	B b1(7);
	a1=b1;
//	a1=(A)b1;
	a1.display();
	getch();
}
```

Q.36 Write a program to show the use of friend function in C++.

```
//Friend function
#include<iostream.h>
#include<conio.h>
class B;
class A
{
        private:
        int x;
        public:
        A()
        {
                x=0;
        }
        A(int x1)
        {
                x=x1;
        }
        friend int sum(A,B);
};
class B
{
        private:
        int y;
        public:
        B()
        {
                y=0;
        }
        B(int y1)
        {
                y=y1;
        }
        friend int sum(A,B);
};
int sum(A a1,B b1)
{
        int c;
        c=a1.x+b1.y;
        return c;
}
int main (void)
{
        A a1(5);
        B b1(6);
        cout<<sum(a1,b1);
        getch();
        return 0;
}
```

Q.37 Write a program to show the use of friend class in C++.

```
//Friend class
#include<iostream.h>
#include<conio.h>
class complex
{
        int real,imag;
        public:
        complex()
        {
                real=imag=0;
        }
        complex(int r,int i)
        {
                real=r;
                imag=i;
        }
        complex operator + (complex c)
        {
                complex t;
                t.real=real+c.real;
                t.imag=imag+c.imag;
                return t;
        }
        friend ostream & operator<<(ostream &,complex &);
        friend istream & operator>>(istream &,complex&);
};
ostream & operator<<(ostream & out,complex & c)
{
        out<<c.real<<"\t"<<c.imag<<endl;
        return out;
}
istream & operator>>(istream & in, complex & c)
{
        cout<<"Enter real ";
        in>>c.real;
        cout<<"Enter imag ";
        in>>c.imag;
        return in;
}
```

```
int main(void)
{
	clrscr();
	complex c1,c2,c3;
	cout<<"Enter two complex numbers: ";
	cin>>c1>>c2;
	c3=c1+c2;
	cout<<"sum = "<<c3;
	getch();
	return 0;
}
```

Q.40 Write a program to overload the relational and equal operator in C++.

```
//relational and equal operator overloading
#include<iostream.h>
#include<conio.h>
#include<string.h>
class string
{
	char str[50];
	public:
	string()
	{
		strcpy(str,"");
	}
	string(char s[])
	{
		strcpy(str,s);
	}
	void getdata()
	{
		cout<<"Enter string";
		cin>>str;
	}
	void display()
	{
		cout<<str<<endl;
	}
	string operator +(string s)
	{
		string t;
		strcpy(t.str,str);
		strcat(t.str,s.str);
		return(t);
	}
```

```
        int operator>(string s)
        {
                return(strcmp(str,s.str)>0);
        }
        int operator<(string s)
        {
                return(strcmp(str,s.str)<0);
        }
        int operator==(string s)
        {
                return((str,s.str)==0);
        }
};
int main(void)
{
        clrscr();
        string s1("mat");
        string s2("rix");
        string s3;
        s3=s1+s2;
        s3.display();
        cout<<endl;
        s1.getdata();
        s2.getdata();
        if(s1>s2)
                cout<<"First is greater";
        else if(s1<s2)
                cout<<"Second is greater";
        else if(s1==s2)
                cout<<"equal strings";
        getch();
        return 0;
}
```

Q.41 Write a program to show single inheritance in C++.

```
//single inheritance
#include<iostream.h>
#include<conio.h>
class A
{
        int x;
        public:
        A()
        {
                x=0;
        }
        A(int x1)
        {
                x=x1;
        }
        void getdata()
        {
                cout<<"enter a number: ";
                cin>>x;
        }
        void display()
        {
                cout<<x<<endl;
        }
};
class B:public A
{
        int y;
        public:
        B():A()
        {
                y=0;
        }
        B(int x1,int y1):A(x1)
        {
                y=y1;
        }
        void getdata()
        {
                A::getdata();
                cout<<"Enter a number: ";
                cin>>y;
        }
        void display()
        {
                A::display();
                cout<<y<<endl;
        }
};
int main()
{
        B b1;
        b1.getdata();
        b1.display();
        B b2(5,7);
        b2.display();
```

```
	getch();
	return 0;
}
```

Q.42 Write a program to show multi level inheritance in C++

```
//multi level inheritence
#include<iostream.h>
#include<conio.h>
class A
{
	int x;
	public:
	A()
	{
		x=0;
	}
	A(int x1)
	{
		x=x1;
	}
	void getdata()
	{
		cout<<"enter a number: ";
		cin>>x;
	}
	void display()
	{
		cout<<x<<endl;
	}
};
class B:public A
{
	int y;
	public:
	B():A()
	{
		y=0;
	}
	B(int x1,int y1):A(x1)
	{
		y=y1;
	}
	void getdata()
	{
		A::getdata();
```

```
                cout<<"Enter a number: ";
                cin>>y;
        }
        void display()
        {
                A::display();
                cout<<y<<endl;
        }
};
class C:public B
{       int z;
        public:
        C():B()
        {
                z=0;
        }
        C(int x1,int y1,int z1):B(x1,y1)
        {
                z=z1;
        }
        void getdata()
        {
                B::getdata();
                cout<<"Enter a number: ";
                cin>>z;
        }
        void display()
        {
                B::display();
                cout<<z<<endl;
        }
};
int main()
{
        C c1;
        c1.getdata();
        c1.display();
        C c2(5,6,7);
        c2.display();
        getch();
        return 0;
}
```

Q.43 Write a program to show multiple inheritance in C++

```
//multiple inheritence
```

```
#include<iostream.h>
#include<conio.h>
class A
{
	int x;
	public:
	A()
	{
		x=0;
	}
	A(int x1)
	{
		x=x1;
	}
	void getdata()
	{
		cout<<"enter a number: ";
		cin>>x;
	}
	void display()
	{
		cout<<x<<endl;
	}
};
class B
{
	int y;
	public:
	B()
	{
		y=0;
	}
	B(int y1)
	{
		y=y1;
	}
	void getdata()
	{
		cout<<"enter a number: ";
		cin>>y;
	}
	void display()
	{
		cout<<y<<endl;
	}
};
```

```
class C:public A,public B
{
        int z;
        public:
        C():A(),B()
        {
                z=0;
        }
        C(int x1,int y1,int z1):A(x1),B(y1)
        {
                z=z1;
        }
        void getdata()
        {
                A::getdata();
                B::getdata();
                cout<<"Enter a number: ";
                cin>>z;
        }
        void display()
        {
                A::display();
                B::display();
                cout<<z<<endl;
        }
};
void main()
{
        C c1;
        c1.getdata();
        c1.display();
        C c2(5,6,7);
        c2.display();
        getch();
}
```

Q.44 Write a program to show object slicing in C++

```
//object slicing
#include<iostream.h>
#include<conio.h>
class A
{
        int x;
    public:
        A()
```

```
	{
		x=0;
	}
	A(int x1)
	{
		x=x1;
	}
	void display()
	{
		cout<<x<<endl;
	}
};
class B : public A
{
	int y;
	public:
	B():A()
	{
		y=0;
	}
	B(int x1, int y1):A(x1)
	{
		y=y1;
	}
	void display()
	{
		A::display();
		cout<<y<<endl;
	}
};
void main()
{
	clrscr();
	A a1;
	B b1(5,6);
	a1=b1; //object slicing
	a1.display();
	getch();
}
```

Q.45 [A] Write a program to show the use of virtual function in C++.

```
//Virtual Function
#include<iostream.h>
#include<conio.h>
class shape
```

```
{
   protected:
          float pi;
          shape()
          {
                 pi=3.141f;
          }
};
class circle:public shape
{
          int r;
          float area;
   public:
          void getdata()
          {
                 cout<<"Enter radius";
                 cin>>r;
          }
          void calc_area()
          {
                 area=pi*r*r;
          }
          void display()
          {
                 cout<<"Area of circle is "<<area;
          }
};
class rect:public shape
{
          int l,b;
          float area;
   public:
          void getdata()
          {
                 cout<<"Enter l,b";
                 cin>>l>>b;
          }
          void calc_area()
          {
                 area=l*b;
          }
          void display()
          {
                 cout<<"Area of rectangle is "<<area;
          }
};
```

```
int main(void)
{
        clrscr();
        shape *p;

        circle c1;
        p=&c1;
        p->getdata();  //error because we can call only those functions
        p->area();     //of child class from parent class pointer which
        p->display();  //are also defined in parent class.

        rect r1;
        p=&r1;
        p->getdata();  //error
        p->area();     //error
        p->display();  //error

/*      circle c1;
        rect r1;
        shape *p[2]={&c1,&r1};
        for(int i=0;i<2;i++)
        {
                p[i]->getdata(); //error
                p[i]->area();    //error
                p[i]->display(); //error
        }
*/
        getch();
        return 0;
}
```

Q.45 [B] Write a program to show the use of virtual function in C++.

```
//Virtual Function
#include<iostream.h>
#include<conio.h>
class shape
{
   protected:
        float pi;
        shape()
        {
                pi=3.141f;
        }
   public:
        void getdata()
        {
                cout<<"inside getdata of class shape"<<endl;
        }
        void calc_area()
        {
                cout<<"inside calc_per of class shape"<<endl;
        }
        void display()
        {
                cout<<"inside display of class shape"<<endl;
        }
};
class circle:public shape
{
        int r;
        float area;
   public:
        void getdata()
        {
                cout<<"Enter radius";
                cin>>r;
        }
        void calc_area()
        {
                area=pi*r*r;
        }
        void display()
        {
                cout<<"Area of circle is "<<area<<endl;
        }
};
class rect:public shape
{
        int l,b;
        float area;
   public:
        void getdata()
        {
                cout<<"Enter l,b";
                cin>>l>>b;
        }
        void calc_area()
        {
                area=l*b;
        }
        void display()
        {
                cout<<"Area of rectangle is "<<area<<endl;
        }
};
int main(void)
{
        clrscr();
        shape *p;

        circle c1;
        p=&c1;
        p->getdata();  //no error but due to early binding it will call
        p->calc_area();     //functions of base class not the
        p->display();  //derived class

        rect r1;
        p=&r1;
        p->getdata();  //no error but due to early binding it will call
        p->calc_area();     //functions of base class not the
        p->display();  //derived class

/*      circle c1;
        rect r1;
        shape *p[2]={&c1,&r1};
        for(int i=0;i<2;i++)
        {
                p[i]->getdata();
                p[i]->calc_area();
                p[i]->display();
        }
*/
        getch();
        return 0;
}
```

Q.45 [C] Write a program to show the use of virtual function in C++.

```
//Virtual Function
#include<iostream.h>
#include<conio.h>
class shape
{
   protected:
        float pi;
        shape()
        {
                pi=3.141f;
        }
    public:
        virtual void getdata()
        {
                cout<<"inside getdata of class shape"<<endl;
        }
        virtual void calc_area()
        {
                cout<<"inside calc_per of class shape"<<endl;
        }
        virtual void display()
        {
                cout<<"inside display of class shape"<<endl;
        }
};
class circle:public shape
{
        int r;
        float area;
    public:
        void getdata()
        {
                cout<<"Enter radius";
                cin>>r;
        }
        void calc_area()
        {
                area=pi*r*r;
        }
        void display()
        {
                cout<<"Area of circle is "<<area<<endl;
        }
};
class rect:public shape
{
        int l,b;
        float area;
    public:
        void getdata()
        {
                cout<<"Enter l,b";
                cin>>l>>b;
        }
        void calc_area()
        {
                area=l*b;
        }
        void display()
        {
                cout<<"Area of rectangle is "<<area<<endl;
        }
};
int main(void)
{
        clrscr();
        shape *p;

        circle c1;
        p=&c1;
        p->getdata();  //no error & due to virtual keyword there will be
        p->calc_area();     //late binding so functions of derived class
        p->display();  //will be called

        rect r1;
        p=&r1;
        p->getdata();  //no error & due to virtual keyword there will be
        p->calc_area();     //late binding so functions of derived class
        p->display();  //will be called

/*      circle c1;
        rect r1;
        shape *p[2]={&c1,&r1};
        for(int i=0;i<2;i++)
        {
                p[i]->getdata();
                p[i]->calc_area();
                p[i]->display();
        }
*/
        getch();
        return 0;
}
```

Q.46 Write a program to show the use of pure virtual function in C++.

```
//Pure Virtual Function
#include<iostream.h>
#include<conio.h>
class shape
{
   protected:
          float pi;
          shape()
          {
                  pi=3.141f;
          }
   public:
          virtual void getdata()=0;
          virtual void calc_area()=0;
          virtual void display()=0;
};
class circle:public shape
{
          int r;
          float area;
   public:
          void getdata()
          {
                  cout<<"Enter radius";
                  cin>>r;
          }
          void calc_area()
          {
                  area=pi*r*r;
          }
          void display()
          {
                  cout<<"Area of circle is "<<area<<endl;
          }
};
class rect:public shape
{
          int l,b;
          float area;
   public:
          void getdata()
          {
                  cout<<"Enter l,b";
                  cin>>l>>b;
          }
          void calc_area()
          {
                  area=l*b;
          }
          void display()
          {
                  cout<<"Area of rectangle is "<<area<<endl;
          }
};
int main(void)
{
          clrscr();

          shape *p;

          circle c1;
          p=&c1;
          p->getdata();
          p->calc_area();
          p->display();

          rect r1;
          p=&r1;
          p->getdata();
          p->calc_area();
          p->display();

/*
          circle c1;
          rect r1;
          shape *p[2]={&c1,&r1};
          for(int i=0;i<2;i++)
          {
                  p[i]->getdata();
                  p[i]->calc_area();
                  p[i]->display();
          }
*/
          getch();
          return 0;
}
```

Q.47 Write a program to show the use of virtual destructor in C++.

```
//Virtual Destructor is possible (Virtual constructor is not possible)
#include<iostream.h>
#include<conio.h>
class A
{
   public:
        A()
        {
                cout<<"Constructor of A"<<endl;
        }
        virtual ~A()
        {
                cout<<"Destructor of A"<<endl;
        }
};
class B:public A
{
    public:
        B()
        {
                cout<<"Constructor of B"<<endl;
        }
        ~B()
        {
                cout<<"Destructor of B"<<endl;
        }
};
int main(void)
{
        clrscr();
        A *p;
        p=new B;
        delete p;
        getch();
        return 0;
}
```

Q.48 Write a program to copy a text file into another file.

```
//copy a text file into another file
#include<fstream.h>
#include<conio.h>
#include<process.h>
int main()
{
        clrscr();
        char ch;
        ifstream fin("a.txt");
        ofstream fout("b.txt");
        if(!fin||!fout)
        {
                cerr<<"file opening error";
                getch();
                exit(1);
```

```
	}
	while(1)
	{
		fin>>ch;		//ch=fin.get();
		if(fin.eof())
			break;
		fout<<ch;		//fout.put(ch);
	}
	fin.close();
	fout.close();
	return 0;
}
```

Q.49 Write a program to enter information of students using text file in C++.

```
#include<fstream.h>
#include<conio.h>
#include<process.h>
#include<string.h>
class student
{
	int roll;
	char name[10];
	public:
	student()
	{
		roll=0;
		strcpy(name,"");
	}
	void getdata()
	{
		cout<<"enter roll number: ";
		cin>>roll;
		cout<<"enter name: ";
		cin>>name;
	}
	void writedisk()
	{
		ofstream fout("student.txt",ios::ate);
		if(!fout)
		{
			cerr<<"file opening error";
			getch();
			exit(1);
		}
		fout<<roll<<"\t"<<name<<endl;
		fout.close();
	}
	void readall();
};
void student::readall()
{
	ifstream fin;
	fin.open("student.txt");
	if(!fin)
	{
		cerr<<"file opening error";
		getch();
		exit(1);
	}
	while(1)
	{
		fin>>roll>>name;
		if(fin.eof())
		{
			break;
		}
		cout<<roll<<"\t"<<name<<endl;
	}
	fin.close();
}
void main()
{
	student s1;
	int n,i;
	cout<<"enter how many students: ";
	cin>>n;
	for(i=1;i<=n;i++)
	{
		s1.getdata();
		s1.writedisk();
	}
	s1.readall();
	getch();
}
```

Q.50 Write a program to change information of students using text file in C++.

```
#include<fstream.h>
#include<conio.h>
#include<process.h>
#include<string.h>
```

```
class student
{
	int roll;
	char name[10];
	public:
	student()
	{
		roll=0;
		strcpy(name,"");
	}
	void getdata()
	{
		cout<<"enter roll number: ";
		cin>>roll;
		cout<<"enter name: ";
		cin>>name;
	}
	void writedisk()
	{
		ofstream fout("student.bin",ios::ate|ios::binary);
		if(!fout)
		{
			cerr<<"file opening error";
			getch();
			exit(1);
		}
		fout.write((char *)this,sizeof(student));
		fout.close();
	}
	int student::count()
	{
		ifstream fin("student.bin",ios::binary);
		fin.seekg(0,ios::end);
		int filesz=fin.tellg();
		int n=filesz/sizeof(student);
		return n;
	}
	void student::change(int rec_no)
	{
		fstream file("student.bin",ios::in|ios::out|ios::binary);
		if(!file)
		{
			cerr<<"file opening error";
			getch();
			exit(1);
		}
```

```
                int pos=((rec_no-1)*sizeof(student));
                file.seekg(pos,ios::beg);
                file.read((char *)this,sizeof(student));
                cout<<"old record"<<endl<<roll<<"\t"<<name<<endl;
                cout<<"Enter new roll number: ";
                cin>>roll;
                cout<<"Enter new name: ";
                cin>>name;
                file.seekp(pos,ios::beg);
                file.write((char *)this,sizeof(student));
                file.close();
        }
        void readall();
};
void student::readall()
{
        ifstream fin;
        fin.open("student.bin",ios::binary);
        if(!fin)
        {
                cerr<<"file opening error";
                getch();
                exit(1);
        }
        while(1)
        {
                fin.read((char *)this,sizeof(student));
                if(fin.eof())
                {
                        break;
                }
                cout<<roll<<"\t"<<name<<endl;
        }
        fin.close();
}
void main()
{
        student s1;
        int n,i,no;
        cout<<"enter how many students: ";
        cin>>n;
        for(i=1;i<=n;i++)
        {
                s1.getdata();
                s1.writedisk();
        }
        s1.readall();
        int recd_no;
        recd_no=s1.count();
        cout<<"Total Records="<<recd_no<<endl;
        cout<<"Enter record number to change";
        cin>>no;
        s1.change(no);
        s1.readall();
        getch();
}
```

Q.51 Write a program to merge two text files in C++.

```
//merge two text files
#include<fstream.h>
#include<process.h>
#include<conio.h>
int main()
{
        clrscr();
        char grade,name[20];
        ifstream fin1("name.txt");
        ifstream fin2("grade.txt");
        ofstream fout("student.txt",ios::ate);
        if(!fin1||!fin2||!fout)
        {
                cerr<<"file opening error";
                getch();
                exit(1);
        }
        while(1)
        {
                fin2>>grade;
                fin1.getline(name,20);
                if(fin1.eof()||fin2.eof())
                        break;
                fout<<name<<"\t"<<grade<<endl;
        }
        fin1.close();
        fin2.close();
        fout.close();
        getch();
        return 0;
}
```

Q.52 Write a program to show the use of template in C++.

```
#include<iostream.h>
#include<conio.h>
template <class T>
T sum(T a, T b)
{
        T  c;
        c = a + b;
        return c;
}
int main(void)
{
        clrscr();
        int a1,b1,ans1;
        cout<<"Enter 2 integer numbers";
        cin>>a1>>b1;
        ans1=sum(a1,b1);
        cout<<"Result is "<<ans1<<endl;
        float a2,b2,ans2;
        cout<<"Enter 2 floating numbers";
        cin>>a2>>b2;
        ans2=sum(a2,b2);
        cout<<"Result is "<<ans2<<endl;
        float a3,b3,ans3;
        cout<<"Enter 2 integer numbers";
        cin>>a3>>b3;
        ans3=sum(a3,b3);
        cout<<"Result is "<<ans3<<endl;
        getch();
        return 0;
}
```

Q.53 Write a program to show template function override in C++.

```
//Template function override
#include<iostream.h>
#include<conio.h>
template <class T>
T mod(T a, T b)
{
        T  c;
        c = a % b;
        return c;
}
int mod(float a, float b)
```

```
{
        int c;
        c=int(a)%int(b);
        return c;
}
int main(void)
{
        clrscr();
        cout<<mod(5,2)<<endl;
        cout<<mod(75000l,50000l)<<endl;
        cout<<mod(5.0f,2.0f)<<endl;
        getch();
        return 0;
}
```

Q.54 Write a program to show the use of multiple template function in C++.

```
//Multiple Template function
#include<iostream.h>
#include<conio.h>
template <class T1,class T2>
T2 div(T1 a, T2 b)
{
        T2  c;
        c = a / b;
        return c;
}
int main(void)
{
        clrscr();
        cout<<div(5,2.0f)<<endl;
        cout<<div(5.0f,2)<<endl;
        getch();
        return 0;
}
```

Q.55 Write a program to show stack using template class in C++.

```
//template class
#include<iostream.h>
#include<conio.h>
#define MAXSTK 10
template <class T>
class Stack
{
        T data[MAXSTK];
        int top;
    public:
        Stack()
        {
                top=-1;
        }
        void push(T item)
        {
                if(top==MAXSTK-1)
                {
                        cerr<<"Overflow";
                        return;
                }
                top++;
                data[top]=item;
        }
        T pop()
        {
                T item;
                if(top == -1)
                {
                        cerr<<"Underflow";
                        return item;
                }
                item=data[top];
                top--;
                return item;
        }
};
int main(void)
{
        clrscr();
        Stack<int> s1;
        s1.push(10);
        s1.push(20);
        s1.push(30);
        cout<<s1.pop()<<endl;
        cout<<s1.pop()<<endl;
        cout<<s1.pop()<<endl;

        Stack<float> s2;
        s2.push(5.4f);
        s2.push(6.3f);
        s2.push(7.2f);
        cout<<s2.pop()<<endl;
        cout<<s2.pop()<<endl;
        cout<<s2.pop()<<endl;
        getch();
        return 0;
}
```

Q.56 Write a program to show template class and non template type in C++.

```
//template class & Non template Type
#include<iostream.h>
#include<conio.h>
template <class T, int MAXSTK>
class Stack
{
        T data[MAXSTK];
        int top;
    public:
        Stack()
        {
                top=-1;
        }
        void push(T);
        T pop();
};
template <class T, int MAXSTK>
void Stack<T,MAXSTK>::push(T item)
{
        if(top==MAXSTK-1)
        {
                cerr<<"Overflow";
                return;
        }
        top++;
        data[top]=item;
}
template <class T, int MAXSTK>
T Stack<T,MAXSTK>::pop()
{
        T item;
        if(top == -1)
        {
                cerr<<"Underflow";
                return item;
        }
        item=data[top];
        top--;
        return item;
}
int main(void)
{
        clrscr();
        Stack<int,10> s1;
        s1.push(10);
        s1.push(20);
        s1.push(30);
        cout<<s1.pop()<<endl;
        cout<<s1.pop()<<endl;
        cout<<s1.pop()<<endl;

        Stack<float,20> s2;
        s2.push(5.4f);
        s2.push(6.3f);
        s2.push(7.2f);
        cout<<s2.pop()<<endl;
        cout<<s2.pop()<<endl;
        cout<<s2.pop()<<endl;
        getch();
        return 0;
}
```

Q.57 Write a program to show nesting of classes in C++.

```
#include <iostream.h>
#include<conio.h>
class A
{
    public:
        int x;
        A()
        {
                x=0;
        }
        void f1()
        {
                B b1;
                b1.f2();
                cout<<"outer class " << b1.y<<endl;
        }
        class B
        {
           public:
                int y;
                B()
```

```
		{
			y=0;
		}
		void f2()
		{
			A a1;
			cout<<"inner class "<<x<<endl;
		}
	};
};
int main(void)
{
	clrscr();
	A a1;
	a1.f1();
	A::B b1;
	b1.f2();
	getch();
	return 0;
}


/*Input / Output flags
1.	Unformatted Input/Output Operations
	a.	Overloaded Operators >> and <<
	b.	put() and get() functions
	c.	getline() and write() functions
2.	Formatted Input/Output Operations
	a.	ios class functions and flags
			width()
			precision()
			fill()
			setf()  (showbase, showpos, showpoint, uppercase, adjustfield,
floatfield, basefield)
			unsetf()
	b.	Manipulators
			setw()
			setprecision()
			setfill()
			setiosflags()
			resetiosflags()
	c.	User defined output functions
*/
```

Q.58 Write a program to show the use of input and output flags in C++.

```
#include<iostream.h>
#include<conio.h>
int main(void)
{
        clrscr();
        int i=52;
        float a=0.00123;
        char str[] = "dream, they make it happen";
        cout.width(4);
        cout<<25<<endl;                     //__25
        cout<<125<<endl;            //125
        cout.width(4);
        cout<<125<<endl;            //_125

        cout.fill('0');
        cout.width(4);
        cout<<25<<endl;                     //0025

        cout<<i<<endl;                      //52
        cout.setf(ios::showpos);
        cout<<i<<endl;                      // +52
        cout.unsetf(ios::showpos);
        cout<<i<<endl;                      // 52

        cout.setf(ios::showbase);
        cout.setf(ios::uppercase);
        cout.setf(ios::hex, ios::basefield);
        cout<<i<<endl;                      //0X34
        cout.setf(ios::oct, ios::basefield);
        cout<<i<<endl;                      //064

        cout.fill('0');
        cout.setf(ios::showpos);
        cout.setf(ios::dec, ios::basefield);
        cout.width(10);
        cout<<i<<endl;                      //0000000+52
        cout.setf(ios::left, ios::adjustfield );
        cout.width(10);
        cout<<i<<endl;                      //+520000000
        cout.setf(ios::internal, ios::adjustfield );
        cout.width(10);
        cout<<i<<endl;                      //+000000052
        cout<<i<<endl;                      //+52
        cout<<5.40<<endl;           //+5.4
        cout<<5.00<<endl;           //+5
        cout.unsetf(ios::showpos);
        cout<<a<<endl;                      //0.00123
        cout.setf(ios::fixed,ios::floatfield);
        cout.precision(6);
        cout.setf(ios::showpoint);
        cout<<5.40<<endl;           //5.400000
        cout<<5.00<<endl;           //5.000000
        cout<<5.3<<endl;            //5.300000
        cout<<a<<endl;                      //0.001230
        cout.setf(ios::scientific,ios::floatfield);
        cout<<a<<endl;                      //1.230000e-002

        cout.width(40);
        cout<<str;          //00000000000000dream, they make it happen
        getch();
        return 0;
}
```

Q.59 Write a program to show the use of pointer to a member variable in C++.

```
#include <iostream.h>
#include<conio.h>
class A
{
        int x,y;
    public:
        void setdata(int a,int b)
        {
                x = a;
                y = b;
        }
        friend int sum(A);
};
int sum(A obj)
{
        int A::*px = &A::x;                     //Pointer to a member variable.
        int A::*py = &A::y;                     //Pointer to a member variable.
        A * pobj = &obj;                        //Pointer to an object.
        int s = obj.*px + pobj->*py;            //Dereferencing pointers
        return s;
}
int main(void)
{
        clrscr();
        A a1;
        void (A::*pf)(int, int) = &A::setdata; //Pointer to a member function.
        (a1.*pf)(10,20);                        //Calling member function.
        cout << "sum = "<<sum(a1) << endl;
        A *op = &a1;                            //Pointer to an object.
        (op->*pf)(30,40);                       //Calling member function.
        cout << "sum= "<< sum(a1) <<endl;
        getch();
        return 0;
}
```